

1 This application is submitted in the name of the following inventor:

2
3 *Inventor* *Citizenship* *Residence City and State*
4 Subir VARMA United States San Jose, CA

5
6 The assignee is *Aperto Networks, Inc.*, a corporation having an office at 1637
7 South Main Street, Milpitas CA 95035.

8
9 TITLE OF THE INVENTION

10
11 Automatic Retransmission and Error Recovery for Packet Oriented Point-To-Multipoint Com-
12 munication

13
14 BACKGROUND OF THE INVENTION

15
16 A portion of the disclosure of this patent document contains material which is
17 subject to copyright protection. The copyright owner has no objection to the facsimile reproduc-
18 tion by anyone of the patent document or the patent disclosure, as it appears in the Patent and
19 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

1 1. *Field of the Invention*

2
3 This invention relates to wireless communication systems, such as those including
4 automatic retransmission and error recovery for packet oriented point-to-multipoint communica-
5 tion.
6

7 2. *Related Art*

8
9 In communication systems, messages from a sender to a receiver using a commu-
10 nication link are sometimes subject to sending errors, such as bit errors, unreasonable sending
11 delay, unintended reordering, and unintended duplication of messages. For example, noise on
12 the communication link can cause bits within messages to be incorrect, generally causing the re-
13 ceiver to be unable to use the message. In a wireless communication system, these problems are
14 exacerbated by a variety of circumstances that are specific to wireless communication. For ex-
15 ample, co-channel interference (CCI), multipath and multipoint effects, such as refraction or re-
16 flection resulting in intrasymbol interference and intersymbol interference, are often prevalent
17 with wireless communication, and can substantially reduce the reliability of wireless communi-
18 cation links.
19

20 One known method in digital communication systems is to implement an auto-
21 matic retransmission protocol between sender and receiver, so that the receiver acknowledges
22 messages from the sender, and the sender re-transmits those messages not acknowledged by the
23 receiver within a reasonable time. Known automatic retransmission protocols include several
24 parameters, which must generally be selected in response to characteristics of the communication
25 link, to optimize communication throughput between the sender and receiver.

1
2 One problem with application of this known method to wireless communication
3 systems is that there are multiple physical characteristics of the wireless communication link,
4 each which is specific to a particular combination of sender and receiver, and each of which can
5 change substantially over relatively short time durations. These multiple physical characteristics
6 can include characteristics of the sender's equipment or of the receiver's equipment, characteris-
7 tics of objects on or near communication pathways between the sender and the receiver, and
8 characteristics of other communications overlapping communication between the sender and the
9 receiver. For example, the wireless communication environment can include substantial changes
10 in wireless communication link characteristics in the time duration between sending a message
11 and sending an appropriate acknowledgement for that message. This is particularly so for char-
12 acteristics related to errors in sending information using wireless communication links, including
13 interference such as CCI, and multipath and multipoint effects. Moreover, multiple ones of these
14 physical characteristics can change independently of one another, and can have substantial and
15 relatively unpredictable effects on one another.

16
17 Accordingly, selection of a single set of parameters with which to optimize auto-
18 matic retransmission using a wireless communication link is virtually always suboptimal for
19 communication among multiple senders and multiple receivers. Moreover, selection of parame-
20 ters with which to optimize automatic retransmission can be subject to substantial data collection
21 and computation; this task is not easily distributed among multiple senders and multiple receiv-
22 ers. Accordingly, it would be advantageous to provide a technique for automatic retransmission
23 and error recovery for packet oriented point to multipoint communication, that is not subject to
24 drawbacks of the known art. Preferably, in such a technique, automatic retransmission and error

1 recovery characteristics are responsive to changes in the characteristics of the communication
2 link between sender and receiver.

4 SUMMARY OF THE INVENTION

6 The invention provides a method and system for point to multipoint wireless
7 communication, including automatic retransmission and error recovery for packet oriented point
8 to multipoint communication. The method and system integrates adaptive and dynamic respon-
9 siveness for parameters for automatic retransmission using wireless communication, both for
10 single sender and a single receiver, and for sets of multiple senders and multiple receivers.

12 In a first aspect of the invention, the wireless communication link is divided into a
13 downstream portion and an upstream portion. The method and system selects parameters for
14 automatic retransmission independently for the downstream portion and the upstream portion of
15 the wireless communication link. A base station controller (BSC) controls the selection of pa-
16 rameters for automatic retransmission for all customer premises equipment (CPE) within a cell.
17 As part of a TDD frame, in which the BSC and the CPE share communication bandwidth using a
18 TDMA technique, the BSC includes its selection of parameters for automatic retransmission to
19 be used by CPE within a control section of the TDD frame.

20 Preferably in this aspect of the invention, the BSC dynamically and adaptively
21 determines new selected parameters for automatic retransmission, in response to conditions of a
22 wireless communication link with each independent CPE. One problem particular to this aspect
23 of the invention, and overcome by the invention, is that when the BSC sends new selected pa-
24 rameters for using the wireless communication link, aspects of each message to be sent will also
25 dynamically vary. These can include the size of each message (in bytes or message symbols), the

1 time duration for each message, and other aspects of each message. Accordingly, in the second
2 aspect of the invention, parameters for automatic retransmission are responsive to a number of
3 bytes successfully sent from a sender to a receiver, rather than responsive to a number of mes-
4 sages successfully sent or a number of symbols successfully sent.

5
6 In a second aspect of the invention, upstream retransmission control is placed in
7 the receiver of the upstream communication (i.e., the BSC) rather than the transmitter of the up-
8 stream communication (i.e., the CPE). In order to control retransmission, the BSC dynamically
9 and adaptively allocates acknowledgement time slots within the upstream portion of the TDD
10 frame for use by each selected CPE. Thus, the BSC, in addition to determining parameters for
11 automatic retransmission, also determines an amount of bandwidth allocated to each selected
12 CPE for sending messages associated with automatic retransmission (such as acknowledgement
13 or non-acknowledgement messages). As part of this third aspect of the invention, the BSC allo-
14 cates some portion of the upstream bandwidth as a shared resource and some portion of the up-
15 stream bandwidth as unshared (that is, specifically allocated to a selected CPE) when there are
16 messages received but not yet acknowledged.

17 In a third aspect of the invention, the BSC dynamically and adaptively responds to
18 acknowledgement and non-acknowledgement messages from each selected CPE, to integrate the
19 automatic retransmission protocol with the TDD frame and the TDMA technique used within
20 that frame. In a preferred embodiment, when the BSC sends messages to a selected CPE, the
21 BSC sets a first timeout each time it receives a non-acknowledgement message from that selected
22 CPE; during this first timeout duration, the BSC discards further acknowledgement and non-
23 acknowledgement messages from that selected CPE. Also in a preferred embodiment, when the
24 BSC receives messages from a selected CPE, the BSC sets a second timeout each time it receives

1 an invalid message from the selected CPE; during this second timeout duration, the BSC discards
2 all further messages received from that selected CPE.

3
4 The invention provides an enabling technology for a wide variety of applications
5 for communication, so as to obtain substantial advantages and capabilities that are novel and
6 non-obvious in view of the known art. Examples described below primarily relate to wireless
7 communication systems, but the invention is broadly applicable to many different types of com-
8 munication in which characteristics of the communication link are subject to change.

9 10 BRIEF DESCRIPTION OF THE DRAWINGS

11
12 Figure 1 shows a block diagram of a portion of a system using automatic retrans-
13 mission and error recovery in a point to multipoint wireless communication.

14
15 Figure 2 shows a time division duplex frame used in a system as in Fig. 1.

16
17 Figure 3 shows a process flow diagram of a method for operating a system as in
18 Fig. 1.

1 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

2
3 In the following description, a preferred embodiment of the invention is described
4 with regard to preferred process steps and data structures. Embodiments of the invention can be
5 implemented using general-purpose processors or special purpose processors operating under
6 program control, or other circuits, adapted to particular process steps and data structures de-
7 scribed herein. Implementation of the process steps and data structures described herein would
8 not require undue experimentation or further invention.

9
10 *Related Applications*

11
12 Inventions described herein can be used in conjunction with inventions described
13 in the following documents.

14
15 U.S. Patent Application Serial No. 09/475,642, Express Mail Mailing No.
16 EL524780018US, filed December 30, 1999 in the names of Reza Majidi-Ahy, Subir
17 Varma, Khuong Ngo, Jean Fuentes and Paul Trong, attorney docket number 164.1002.01,
18 titled "Adaptive Link Layer for Point to Multipoint Communication System."

19
20 U.S. Patent Application Serial No. 09/475,716, Express Mail Mailing No.
21 EL524780021US, filed December 30, 1999 in the names of Reza Majidi-Ahy, Joseph
22 Hakim, and Subir Varma, attorney docket number 164.1003.01, titled "Integrated Self-
23 Optimizing Multi-Parameter and Multi-Variable Point to Multipoint Communication
24 System."

1 U.S. Patent Application Serial No. 09/540,674, Express Mail Mailing No.
2 EL524781512US, filed March 31, 2000, in the name of Reza Majidi-Ahy, attorney docket
3 number 164.1001.01, titled "Robust Topology Wireless Communication Using
4 Broadband Access Points."

5
6 U.S. Patent Application Serial No. 09/604,784, Express Mail Mailing No.
7 EL524781225US, filed June 26, 2000 in the names of Reza Majidi-Ahy, attorney docket
8 number 164.1010.01, titled "High-Capacity Scalable Integrated Wireless Backhaul for
9 Broadband Access Networks."

10
11 and

12 U.S. Patent Application Serial No. 09/475,716, Express Mail Mailing No.
13 EL524780021US, filed December 30, 1999 in the name of Reza Majidi-Ahy, Joseph Ha-
14 kim, and Subir Varma, attorney docket number 164.1003.01, titled "Integrated, Self-
15 Optimizing, Multi-Parameter/Multi-Variable Point-to-Multipoint Communication System
16 [II]."

17 Each of these documents is hereby incorporated by reference as if fully set forth herein.
18 This application claims priority of each of these documents. These documents are collectively
19 referred to as the "Incorporated Disclosures."

20
21 *Lexicography*

22
23 The following terms refer or relate to aspects of the invention as described below. The
24 descriptions of general meanings of these terms are not intended to be limiting, only illustrative.

1 **base station controller (BSC)** — in general, a device for performing coordination and
2 control for a wireless communication cell. There is no particular requirement that the base
3 station controller must be a single device; in alternative embodiments, the base station
4 controller can include a portion of a single device, a combination of multiple devices, or
5 some hybrid thereof.

6
7 **communication link** — in general, an element for sending information from a sender to a
8 recipient. Although in a preferred embodiment the communication links referred to are
9 generally wireless line of sight point to point communication links, there is no particular
10 requirement that they are so restricted.

11
12 **customer premises equipment (CPE)** — in general, a device for performing communi-
13 cation processes and tasks at a customer location, and operating in conjunction with the
14 base station controller within a wireless communication cell. There is no particular re-
15 quirement that the customer premises equipment must be a single device; in alternative
16 embodiments, the customer premises equipment can include a portion of a single device,
17 a combination of multiple devices, or some hybrid thereof.

18
19 **IP parameters** — in general, a set of characteristics or parameters relating to an IP layer
20 for a communication link.

21
22 **media-access-control (MAC) parameters** — in general, with reference to a wireless
23 communication link, a set of characteristics or parameters relating to media access control
24 of a communication link. For example, MAC parameters can include (a) a number of
25 payload data bytes assigned per message, (b) a frequency of acknowledgement messages

1 and a number of message retransmission attempts, (c) a fraction of the communication
2 link allocated to downstream versus upstream communication, and the like.

3
4 **physical (PHY) parameters** — in general, with reference to a wireless communication
5 link, a set of characteristics or parameters relating to physical transmission of information
6 on a communication link. For example, physical characteristics can include (a) a symbol
7 transmission rate, (b) a number of payload data bits assigned per symbol, (c) a number of
8 error detection or correction bits assigned per symbol, and the like.

9
10 **QoS parameters** — in general, a set of characteristics or parameters relating to QoS
11 (quality of service) for a communication link.

12
13 **wireless communication system** — in general, a communication system including at
14 least one communication link that uses wireless communication techniques.

15
16 **wireless transport layer** — in general, a set of protocols and protocol parameters for
17 sending and receiving information using wireless transport. In a preferred embodiment,
18 the wireless transport layer is part of a multilayer systems architecture, in which the
19 wireless transport layer is built using a physical transport layer, and the wireless transport
20 layer is used by a logical transport layer such as IP.

21
22 As noted above, these descriptions of general meanings of these terms are not in-
23 tended to be limiting, only illustrative. Other and further applications of the invention, including
24 extensions of these terms and concepts, would be clear to those of ordinary skill in the art after
25 perusing this application. These other and further applications are part of the scope and spirit of

1 the invention, and would be clear to those of ordinary skill in the art, without further invention or
2 undue experimentation.

3
4 *System Context*

5
6 The context of the invention is similar to that of the Incorporated Disclosures.

7 A system using adaptive point to multipoint wireless communication in a wireless
8 communication system operates as part of a system in which devices coupled to a network (such
9 as a computer network) send messages, route and switch messages, and receive messages. In a
10 preferred embodiment, devices coupled to (and integrated with) the network send, route, and re-
11 ceive these messages as sequences of packets, each of which has a header including delivery in-
12 formation and a payload including data. In a preferred embodiment, packet format conforms to
13 the OSI model, in which an application protocol (layer 5, such as FTP), uses a transport protocol
14 (layer 4, such as TCP), which uses a network protocol (layer 3, such as IP), which uses a media
15 access control (MAC) protocol (layer 2), which uses a physical transport technique (layer 1).

16
17 The system using adaptive point to multipoint wireless communication is de-
18 scribed herein with regard to layer 1 and layer 2, particularly as it applies to interactions between
19 layer 1 and layer 2 and between those layers and layer 3. However, concepts and techniques of
20 the invention are also applicable to other layers of the OSI model. The application gives exam-
21 ples of cases where the type of application in the application layer (layer 5) could be incorporated
22 into embodiments of the invention to improve communication. Adapting those concepts and
23 techniques to such other layers would not require undue experimentation or further invention,
24 and is within the scope and spirit of the invention.

1 *System Elements*

2
3 Fig. 1 shows a block diagram of a portion of a system using automatic retrans-
4 mission and error recovery in a point to multipoint wireless communication.

5
6 A system 100 includes a wireless communication cell 110 (or a portion thereof), a
7 base station controller (BSC) 120, one or more customer premises equipment (CPE) 130, and one
8 or more (possibly partially) interfering or reflecting obstacles 140.

9
10 The wireless communication cell 110 includes a generally hexagon-shaped region
11 of local surface area, such as might be found in a metropolitan region. Use of generally hexagon-
12 shaped regions is known in the art of wireless communication because they are able to tile a local
13 region with substantially no gaps. However, although in a preferred embodiment the wireless
14 communication cell 110 includes a generally hexagon-shaped region, there is no particular re-
15 quirement for using that particular shape; in alternative embodiments it may be useful to provide
16 another shape or tiling of the local surface area.

17
18 In Fig. 1, a portion of the cell 110, herein called a “sector” 111, includes a gener-
19 ally triangular-shaped region of local surface area, disposed so that a set of six sectors 111 are
20 combined to form a single cell 110. Thus, the BSC 120 is disposed at or near one corner of the
21 sector 111, while CPE 130 are disposed within the sector 111. Moreover, obstacles 140 are dis-
22 posed within the sector 111 or at junctions of multiple sectors 111.

23 Although the invention is primarily described with regard to a single sector 111,
24 there are substantial applications of the invention to interaction between multiple sectors 111
25 within a cell 110, and to interaction between sectors 111 in multiple cells 110. These substantial
26 applications of the invention are described at least in part in this application. Moreover, other and

1 further substantial applications of the invention with regard to multiple sectors 111, both within a
2 single cell 110 and among multiple cells 110, would be clear to those skilled in the art of wireless
3 communication after perusal of this application, and would not require undue experimentation or
4 further invention.

5
6 The BSC 120 includes a processor, program and data memory, mass storage, and
7 one or more antennas for sending or receiving information using wireless communication tech-
8 niques.

9
10 Similar to the BSC 120, each CPE 130 includes a processor, program and data
11 memory, mass storage, and one or more antennas for sending or receiving information using
12 wireless communication techniques.

13
14 Obstacles 140 might include buildings, other construction, electromagnetically
15 active elements such as radio transmitters and repeaters, other electromagnetic elements such as
16 power lines or weather effects, and possibly mobile objects such as vehicles.

17
18 Although the invention is primarily described with regard to non-moving obsta-
19 cles 140, it would be clear to those of ordinary skill in the art of wireless communication, after
20 perusal of this application, that even non-moving obstacles 140 might present substantial varia-
21 tion over time in characteristics of communication links between the BSC 120 and selected CPE
22 130. Moreover, there are substantial applications of the invention to cells 110 and sectors 111 in
23 which there are moving obstacles 140. Although these substantial applications of the invention
24 are not described in great detail herein, other and further substantial applications of the invention
25 with regard to moving obstacles 140, both within a single cell 110 and among multiple cells 110,

1 would be clear to those skilled in the art of wireless communication after perusal of this applica-
2 tion, and would not require undue experimentation or further invention.

3
4 Communication among devices within the wireless communication cell 110 is
5 preferably conducted on a one-to-one basis between each CPE 130 and the BSC 120. Thus, the
6 BSC 120 communicates with each CPE 130, and each CPE 130 communicates with the BSC
7 120. In a preferred embodiment, CPE 130 do not communicate directly with other CPE 130.
8 However, in alternative embodiments, CPE 130 may communicate directly with other CPE 130,
9 with the characteristics of such communication being controlled either by the BSC 120, by one
10 CPE 130 selected by the BSC 120, or by one CPE 130 mutually agreed to among the communi-
11 cating CPE 130.

12
13 Communication between the BSC 120 and each CPE 130 is conducted using a
14 TDD technique, in which time durations are divided into repeated individual frames, each one of
15 which includes a “downstream” portion and an “upstream” portion. Unlike existing protocols in
16 which transmissions are controlled by the transmitting side, the BSC 120 controls transmissions
17 for both upstream and downstream directions, without specific requests from CPE 130.

18 *Time Division Duplex (TDD) Frame*

19
20 Fig. 2 shows a time division duplex frame used in a system as in Fig. 1.

21
22 During the downstream portion of each frame, the BSC 120 transmits, thus send-
23 ing information to one or more CPE 130. During the upstream portion of each frame, each CPE
24 130 is potentially allocated a time slot for transmission, thus for sending information to the BSC
25 120. TDD techniques are known in the art of wireless communication.

1 A time division duplex (TDD) frame 200 includes a time-synchronization portion
2 210, a first guard time 220, a downstream portion 230, a second guard time 240, a status-
3 synchronization portion 250, and an upstream portion 260.

4
5 The time-synchronization portion 210 includes a first symbol 211 indicating the
6 beginning of the TDD frame 200, and a sequence of parameter setting values 212 for each CPE
7 130. The BSC 120 uses the parameter setting values 212 to inform each selected CPE 130 indi-
8 vidually and separately of (a) the PHY and MAC parameters the BSC 120 is using to send mes-
9 sages to that selected CPE 130, and (b) the PHY and MAC parameters the selected CPE 130
10 should use to send messages to the BSC 120 during its allocated part of the upstream portion
11 260.

12
13 The first guard time 220 includes a time duration sufficient for the BSC 120 to
14 assure that all CPE 130 do not interfere with each other when receiving from the BSC 120 or
15 sending to the BSC 120.

16
17 The downstream portion 230 includes a sequence of downstream payload ele-
18 ments 231, each sent by the BSC 120 to a selected CPE 130. The BSC 120 determines a length
19 for each of these downstream payload elements 231 and sends that information with the parame-
20 ter setting values 212 in the time-synchronization portion 210. In alternative embodiments, the
21 BSC 120 may divide the CPE 130 into classes and allocate one or more downstream payload
22 elements 231 for each class of CPE 130. For example, the BSC 120 may allocate one or more
23 downstream payload elements 231 for broadcast or multicast messages.

1 The second guard time 240 includes a time duration sufficient for the BSC 120 to
2 assure that the downstream portion 230 and the status-synchronization portion 250 do not inter-
3 fere.

4
5 The status-synchronization portion 250 includes a sequence of status information
6 so that the BSC 120 can agree with each selected CPE 130 regarding higher-level protocol status
7 out-of-band from those higher-level protocols.

8
9 Similar to the downstream portion, the upstream portion 260 includes a sequence
10 of upstream payload elements 261, each sent by a selected CPE 130 to the BSC 120. The BSC
11 120 (not the CPE 130) determines a length for each of these upstream payload elements 261 and
12 sends that information with the parameter setting values 212 in the time-synchronization portion
13 210. In alternative embodiments, the BSC 120 may divide the CPE 130 into classes and allocate
14 one or more upstream payload elements 261 for each class of CPE 130, such as for upstream
15 bandwidth contention.

16 17 *Method of Operation*

18
19 Fig. 3 shows a flow diagram of a method for operating a system as in Fig. 1.

20
21 A method 300 includes a set of flow points and a set of steps. The system 100 per-
22 forms the method 300. Although the method 300 is described serially, the steps of the method
23 300 can be performed by separate elements in conjunction or in parallel, whether asynchro-
24 nously, in a pipelined manner, or otherwise. There is no particular requirement that the method
25 300 be performed in the same order in which this description lists the steps, except where so in-
26 dicated.

1
2 At a flow point 310, the BSC 120 and the CPE 130 are ready to begin a TDMA
3 frame.

4
5 At a step 311, the BSC 120 and the CPE 130 conduct communication using a
6 TDMA frame. As part of this step, the BSC 120 directs the CPE 130 regarding which physical
7 parameters and MAC parameters to use.

8
9 At a step 312, the BSC 120 determines characteristics of the communication link
10 with the CPE 130, in response to performance of the communication during the previous TDMA
11 frame.

12
13 At a step 313, the BSC 120 determines exact values for the physical parameters
14 and MAC parameters in response to characteristics of the communication link.

15
16 At a step 314, the BSC 120 determines new values for the physical parameters and
17 MAC parameters for automatic retransmission in response to results of the previous step.

18
19 The BSC 120 preferably determines these automatic retransmission parameters
20 dynamically and adaptively for all CPEs 130 in cell 110. The automatic retransmission parame-
21 ters preferably are determined independently for each upstream portion and each downstream
22 portion and independently for each CPE.

23
24 In order to account for differing characteristics of transmission and retransmission
25 among the CPEs (e.g., message size and duration, symbol size, and other aspects), parameters for

1 retransmission preferably are responsive to a number of bytes successfully transmitted rather than
2 a number of messages or symbols successfully transmitted.

3
4 At step 315, the BSC 120 dynamically and adaptively allocates acknowledgement
5 time slots within upstream portion 260 of each TDD frame for use by the CPEs 130. As part of
6 this step, the BSC 120 preferably allocates some portion of the upstream bandwidth as a shared
7 resource and some portion of the upstream bandwidth as unshared (that is, specifically allocated
8 to selected CPEs) when there are messages received but not yet acknowledged.

9
10 Thus, control of upstream retransmission is placed within the BSC, not the CPE.
11 This control allows the BSC 120 to distribute acknowledgement slots for plural CPEs across plu-
12 ral TDD frames, thereby allowing the BSC 120 to prevent the acknowledgement slots from con-
13 suming too much bandwidth.

14
15 At step S316, the BSC 120 dynamically and adaptively responds to acknow-
16 ledgement and non-acknowledgement messages from each selected CPE 130 so as to integrate
17 the automatic retransmission protocol with the TDD frame and the TDMA technique used within
18 that frame.

19
20 In the preferred embodiment, when the BSC 120 sends messages to a selected
21 CPE 130, the BSC 120 sets a first timeout each time the BSC 120 receives a non-
22 acknowledgement message from that selected CPE. During this first timeout duration, the BSC
23 120 discards further acknowledgement and non-acknowledgement messages from that selected
24 CPE 130.

1 Also in the preferred embodiment, when the BSC 120 receives messages from a
2 selected CPE 130, the BSC 120 sets a second timeout each time it receives an invalid message
3 from the selected CPE 130. During this second timeout duration, the BSC 120 discards all fur-
4 ther messages received from that selected CPE 130.

5
6 After step 316, the BSC 120 and the CPE 130 have performed one step of sending
7 and receiving information using a TDD frame. The flow point 310 is reached repeatedly and the
8 steps thereafter are performed repeatedly, for each TDD frame.

9
10 Pseudo-code for implementing the preferred embodiment of the invention sub-
11 stantially as discussed above is included in a technical appendix to this application.

12 13 *Generality of the Invention*

14
15 The invention has general applicability to various fields of use, not necessarily
16 related to the services described above. For example, these fields of use can include one or more
17 of, or some combination of, the following:

18
19 The invention is applicable to other forms of wireless communication, such as frequency
20 division multiple access (FDMA) or code division multiple access (CDMA, also known
21 as spread spectrum communication);

22
23 The invention is applicable to any non-wireless communication, in which relative effec-
24 tiveness or efficiency of communication can be achieved from dynamically adjusting
25 communication parameters, such as physical parameters or MAC parameters. For exam-

1 ple, the invention can be generalized to non-wireless communication using modems in
2 which equalization parameters are to be dynamically adjusted.

3 The invention is applicable to other wireless communication systems, such as satellite
4 communication systems and (microwave tower or other) point to point transmission sys-
5 tems.

6
7 The invention is applicable to both fixed wireless communication systems, in which cus-
8 tomer premises equipment do not move relative to the BSC 120, and to mobile wireless
9 communication systems, and which customer premises equipment move substantially
10 relative to the BSC 120.

11
12 The invention is applicable to both a single sender and a single receiver, and sets of mul-
13 tiple senders and multiple receivers.

14
15 Other and further applications of the invention in its most general form, will be
16 clear to those skilled in the art after perusal of this application, and are within the scope and spirit
17 of the invention.

18
19 Although preferred embodiments are disclosed herein, many variations are possi-
20 ble which remain within the concept, scope, and spirit of the invention, and these variations
21 would become clear to those skilled in the art after perusal of this application.

TECHNICAL APPENDIX

Pseudo-code copyright 2000 Aperto Networks, Inc.

4.0 Downstream ARQ (BSC Tx, CPE Rx)

4.1 Parameters (Control PDU Handler)

```

ARQWindowSize;    // Size of the ARQ window. Set to 2^(n-1) bytes, where n is the
                  // number of bits in the Sequence Number field
maxAcksLost;       // Maximum number retries for the upstream ACKs, after which
                  // the CPE is re-ranged
maxReqRetries;     // Maximum number of retries for a REQ packet.

```

4.2 BSC Tx (reqWin, scWin, curWin, ackWin)

4.2.1 Initialize (Control PDU Handler)

```

reqWinOff = 0;      // Sequence number of next byte to be queued
scWinOff = 0;       // Sequence number of next byte to be transmitted by BSC
curWinOff = 0;      // Sequence number of next byte the CPE expects
ackWinOff = 0;      // Sequence Number of next byte awaiting acknowledgment
                  // Allocate empty SIDQ_EL and initialize pointers
newSidQEl = AllocateSidQEl();
newSidQEl->EOL = TRUE;
writeElPtr = ackElPtr = curElPtr = scElPtr = newSidQEl;
ackPtr = curPtr = scPtr = 0;
retryCnt = 0;       // Used to decide when to drop a packet
NumAcksLost = 0;    // Used for link adaptation

```

4.2.2 PDU Arrival (Classifier, Policer)

```

// Classify the WPDU
find sidQ (PDU);           // Classifier
// Enqueue the WPDU on the overflow section of the sidQ
newSidQEl = AllocateSidQEl(); // Policer
newSidQEl->EOL = TRUE;        // Policer
writeElPtr->next = newSidQEl; // Policer
writeElPtr->length = PDU.length; // Policer
writeElPtr->txMsgPtr = PDU.txMsgPtr; // Policer
writeElPtr->pktPtr = PDU.packet; // Policer
writeElPtr = newSidQEl;      // Policer
// Traffic shaping may be done before the packet is moved out of the overflow section.
// These updates must be done last to avoid timing problems with USG.
reqWinOff = reqWinOff + PDU.size; // Policer
writeElPtr->EOL = FALSE;          // Policer

```

4.2.3 MAP Construction (Scheduler)

```

while (space left for data in downstream TDD frame) {
    sidQCtrl = SID that Scheduler selects;
    bytesInQueueToSchedule = reqWinOff - scWinOff;
    // Always try to schedule bytes for SIDs without ARQ.
}

```

```

1      // For SIDs with ARQ, we need to make sure that we have not
2      // exhausted our window before we try to schedule some bytes.
3      if ( (sidQCtrl.sidCfgBits.arq = FALSE) OR
4          ((scWinOff + bytesScheduled - ackWinOff) < ARQWindowSize) ) {
5          DATA_GRANT_IE.winOff = scWinOff;
6          DATA_GRANT_IE.payloadSize = bytesScheduled; // Includes delimiter
7      bytes
8          scWinOff = scWinOff + DATA_GRANT_IE.payloadSize;
9          allocate ticks for WPDU in downstream portion of TDD Frame;
10         update scElPtr and scPtr to reflect bytes scheduled;
11         // Mark SID as needing ACK
12         if ( (sidQCtrl.sidCfgBits.ack = TRUE) AND (!sidQCtrl.ackFlag) ) {
13             sidQCtrl.ackFlag = TRUE;
14             add to list of downstream SIDs needing ACK; } } } // while (space
15 left)
16 // Schedule only one ACK per SID for a frame.
17 // We can schedule ACKs for SIDs without ARQ. This is needed for link adaptation.
18 for each SID on list of downstream SIDs needing ACK {
19     // If there are bytes remaining to be acked, allocate space for the
20     // ACK even if the current frame has no WPDUs scheduled for this SID
21     if (scWinOff != ackWinOff) {
22         Allocate ticks for ACK in the upstream portion of TDD frame;
23         ACK_IE.sid = this SID; }
24     else {
25         delete from list of SIDs needing ACK;
26         sidQCtrl.ackFlag = FALSE; }
27 } // for (each SID on list)
28 4.2.4 MAP Arrival (Hardware)
29 if (data grant IE) {
30     // Was a packet dropped or retransmitted?
31     if((sidQCtrl.sidCfgBits.arq=TRUE) AND
32     (curWinOff!=DATA_GRANT_IE.winOff)){
33         // if possible, check (ackWinOff == DATA_GRANT_IE.winOff)
34         curWinOff = ackWinOff;
35         Reset the cur pointers to the ack pointers; }
36     // Need pseudocode for HW packet fragmentation
37     Build a WPDU using the curElPtr and curPtr
38     WPDU.winOff = curWinOff; // Should we use data grant ie not curWin?? JF
39     curWinOff = curWinOff + DATA_GRANT_IE.payload
40     Update curElPtr and curPtr to reflect bytes transmitted }
41 4.2.5 WPDU Transmit (Hardware)
42     transmit built WPDU;
43     if (sidQCtrl.sidCfgBits.arq == FALSE)
44         return any completely transmitted packet;

```

4.2.6 ACK Arrival (Scheduler)

```

1  // Calculate the number of ACKed bytes
2  NumAcksLost = 0;
3  ackByteCnt = ACK.winOff - ackWinOff;
4  // Only free buffers here if ARQ. Otherwise they'd have been freed right after transmit.
5  if (sidQCtrl.sidCfgBits.arq == TRUE) {
6      // Any bytes ACKed?
7      if (ackByteCnt) {
8          ackWinOff = ackWinOff + ackByteCnt;
9          tempElPtr = ackElPtr;
10         update ackPtr and ackElPtr to account for the bytes ACKed;
11         if (tempElPtr != ackElPtr)
12             free SIDQ_ELs between ackElPtr and tempElPtr;
13         if (ACK.nakFlag clear)
14             retryCnt = 0; }
15     // Any bytes NACKed?
16     if (ACK.nakFlag set) {
17         if ((ackByteCnt == 0) && (time > threshold)) {
18             threshold = time at which the last (partially) allocated TDD frame
19             ends;
20             retryCnt = retryCnt + 1; }
21         // When the retry count expires, drop only the first packet in the list.
22         if (retryCnt > sidQCtrl.maxRetry) {
23             // pktPtr points to the first byte in the packet, and ackPtr is the off-
24             set
25             // from pktPtr to the next byte to ack
26             dropBytes = ackElPtr->length - ackPtr;
27             tempElPtr = ackElPtr;
28             update ackElPtr to next packet in list;
29             ackPtr = 0;
30             free (tempElPtr);
31             // Account for any bytes that need to be retransmitted
32             reqWinOff -= dropBytes; // Scheduler asks Policer to do this and
33             does
34             // not schedule any more bytes for this
35             SID
36             // until it is done.
37             inform link adaptation task that we dropped an EPDU }
38         // We have to reschedule some bytes for retransmission
39         scWinOff = ackWinOff;
40         update sc pointers to ack pointers;
41     } // if nakByteCnt
42 } // if ARQ

```

4.2.7 ACK Lost (Scheduler)

```

1      NumAcksLost = NumAcksLost + 1;
2      if (NumAcksLost > maxAcksLost)
3          ReRange CPE;
4      // Note: ACK may be lost if the corresponding MAP was lost. However it is not clear how
5      // a lost MAP event may be detected by the BSC.
6      // Note: If a CPE cannot be ReRanged, the Link Adaptation Task needs to send a message
7      // to the Control PDU Handler to flush the sidQ.
8  4.3 CPE Rx (curWin)
9  4.3.1 Initialize (Control PDU Handler)
10     // CPE S/W does not care about winOffs
11     curWinOff = 0; // Sequence number of the next WPDU to transmit/receive
12     cur pointers = NULL;
13 4.3.2 WPDU Arrival (Hardware)
14     // Never keep bad wpdus
15     if (crc error) {
16         Set NAK flag;
17         Discard(WPDU);
18         discard any packet currently being reassembled; }
19     else if (no energy detected)
20         Set NAK flag;
21     // If an out of sequence wpdu arrives and this SID has ARQ, discard the
22     // wpdu until we receive the next sequence number we are expecting.
23     else if ( (sidQCtrl.sidCfgBits.arq = TRUE) AND (WPDU.winOff != curWinOff) )
24         Discard(WPDU);
25     // Receive the WPDU. Either it's in correct sequence, or the SID has no ARQ and
26     // doesn't care about the sequence.
27     else {
28         curWinOff = WPDU.winOff + WPDU.payloadSize;
29         // Need pseudocode for HW packet reassembly
30         // if a new packet arrives and we were previously assembling a packet,
31         // we discard the old packet and accept the new.
32         if ((WPDU.catPtr == 0) and (curPtr != 0)) {
33             Discard(Partial assembled packet);
34             curPtr = 0;
35             curElPtr = NULL; }
36         // if possible, check the new packet for incorrect length, cuz if it's wrong, and
37         // we don't find it here, it'll be a real bugger to track down
38         if (curElPtr.length != curPtr??)
39             discard packet; }
40 4.3.3 ACK Transmission (Hardware)
41     // When wpdus are scheduled for SIDs with ACK, the Scheduler
42     // will create an IE in the same MAP or in the following MAP
43     // for the ACK.
44     if (sidQCtrl.sidCfgBits.ack == TRUE) {

```



```

1      ACK.status = ACK or NAK;
2      ACK.winOff = curWinOff;
3      ACK.linkParms = modemStatus;
4      Transmit ACK; }
5  5.0 Upstream ARQ (CPE Tx, BSC Rx)
6  5.1 CPE Tx (reqWin, curWin, ackWin)
7  5.1.1 Initialize (Control PDU Handler)
8      reqWinOff = 0; // Sequence Number for the number of the next byte awaiting
9      // transmission.
10     curWinOff = 0; // Sequence Number of the next byte that the CPE expects to tx. The
11     // sequence number in the MAP may be less than this, in case of
12     // re-transmissions.
13     ackWinOff = 0; // Sequence Number of the next byte awaiting acknowledgment.
14     // Allocate empty SIDQ_EL and initialize pointers
15     newSidQEl = AllocateSidQEl();
16     newSidQEl->EOL = TRUE;
17     writeElPtr = ackElPtr = curElPtr = readElPtr = newSidQEl;
18     ackPtr = curPtr = 0;
19  5.1.2 PDU Arrival (Classifier, Policer)
20     // Classify the WPDU
21     find sidQCtrl (PDU); // Classifier
22     // Create new empty SidQEl to terminate list
23     newSidQEl = AllocateSidQEl(); // Policer
24     newSidQEl->EOL = TRUE; // Policer
25     // Enqueue the WPDU on the overflow section of the sidQ. EOL bit should already be set.
26     writeElPtr->next = newSidQEl; // Policer
27     writeElPtr->length = PDU.length; // Policer
28     writeElPtr->txMsgPtr = PDU.txMsgPtr; // Policer
29     writeElPtr->pktPtr = PDU.packet; // Policer
30     writeElPtr = newSidQEl; // Policer
31     if (sidQCtrl->flushFlag not set) {
32         wait till activeFlowFifo has room;
33         activeFlowFifo = PDU.sidNumber; // Policer notifies Hw }
34     // Traffic shaping may be done before the packet is moved out of the overflow section
35     reqWinOff = reqWinOff + PDU.size; // Policer
36     writeElPtr->EOL = FALSE; // Policer
37  5.1.3 REQ Transmission (HW)
38     if (state = Idle) {
39         PDU arrival
40         Compute Defer
41         state = Deferring; }
42     else if (state = Deferring) {
43         map arrives with req IE opportunity
44         REQ.winOff = curWinOff;

```

```

1          REQ.reqWinOff = reqWinOff;
2          Tx REQ;
3          state = GrantPending;
4      else if (state = GrantPending)
5          // The BSC received our REQ packet
6          map arrives with upstream data IE
7      transmit WPDU;
8      // Any more bytes left in SID queue?
9      if (reqWin - curWin)
10         state = GrantPending;
11     // SID queue is empty
12     else {
13         numReqRetries = 0;
14         state = Idle; }
15     // Our REQ packet did not get to the BSC
16     map arrives with no grant IE or grant pending IE
17         numReqRetries = numReqRetries + 1;
18         if (numReqRetries > maxReqRetries) {
19             HW writes SID num plus flush flag in fifo;
20             HW does not tx anymore pdus until sw writes to ACTIVE_SID_FIFO;
21             HW sets sidQCtrl->flushFlag;
22             numReqRetries = 0;
23             state = Idle; }
24         else
25             state = Deferring;
26 5.1.4 MAP Arrival (Hardware)
27     if (MAP missing) {
28         calculate time of next MAP;
29         assume largest MAP size;
30         program Broadcom to receive next MAP; }
31     if (Data Grant IE) {
32         // If ARQ, don't do anything until a grant gives us the expected offset
33         if((sidQCtrl.sidCfgBits.arq == FALSE) OR
34            (DATA_GRANT_IE.winOff == curWinOff)){
35             WPDU.payloadSize = DATA_GRANT_IE.payloadSize;
36             Confirm that allocated ticks are sufficient to accommodate WPDU;
37             WPDU.req = reqWinOff;
38             WPDU.winOff = curWinOff; } }
39     if ((MAP ACK IE) OR (MAP NAK IE)) {
40         if (sidQCtrl.sidCfgBits.arq == TRUE) {
41             ackByteCnt = ACK.winOff - ackWinOff;
42             ackWinOff = ACK.winOff;
43             // Any bytes ACKed?
44             if (ackByteCnt) {

```

```

1          update ackElPtr to account for the ackByteCnt;
2          ackPtr = 0; }
3      if (MAP NAK IE) {
4          reset cur pointers and winOff to ack pointers and winOff; }
5      // Notify SW of ACK, so it can free buffers.
6      write SID number and set ACK flag in the WM_TX_PKT_FIFO; } // if
7  ARQ
8  } // if ACK or NAK IE
9  if (MAP FLUSH IE) {
10     // The Scheduler decided it was time to give up on the packet, so drop the
11     // EOL or End Of List packet.
12     write SID number and set flush flag in WM_TX_PKT_FIFO;
13     set sidQCtrl->flushFlag;
14     // Force data transmission on this SID to halt. This gives us time to
15     // update the reqWinOff.
16     HW does not tx anymore pdus until SW writes to ACTIVE_SID_FIFO;
17     go to req state Idle; }
18

```

5.1.5 Process Tx Pkt Fifo (WMAC Driver)

```

19     read SID number from WM_TX_PKT_FIFO;
20     if (ACK flag) {
21         free SIDQ_Els from readElPtr to ackElPtr;
22         readElPtr = ackElptr; }
23     if (flush flag)
24         send Flush msg to Policer;
25

```

5.1.6 Flush Packet (Policer)

```

26     // Software temporarily has write access to all sidQCtrl fields.
27     drop EOL PDU;
28     update ackElPtr to skip remainder of dropped PDU;
29     ackPtr = 0;
30     curWinOff = ackWinOff;
31     reqWinOff = reqWinOff - remainder of dropped PDU;
32     update curPtr and curElPtr to ackPtr and ackElPtr;
33     clear sidQCtrl->flushFlag;
34     // Kick off another REQ if there are any bytes still on the queue.
35     if (reqWinOff - curWinOff)
36         write SID number to ACTIVE_SID_FIFO;
37

```

5.1.7 WPDU Transmission (Hardware)

```

38     extract WPDU.payloadSize bytes from position curWinOff in SID queue;
39     advance curElPtr and curPtr by WPDU.payloadSize bytes;
40     curWinOff = curWinOff + WPDU.payloadSize;
41     if (sidQCtrl.sidCfgBits.arq == FALSE)
42         return any completely transmitted packet;
43

```

5.2 BSC Rx (reqWin, scWin, curWin)

```

1  5.2.1 Initialize (Control PDU Handler)
2      scWinOff = 0; // Sequence Number of next byte to be transmitted by CPE
3      curWinOff = 0; // Sequence Number of the next byte that the BSC expects
4      reqWinOff = 0; // Cumulative count of number of bytes received at CPE
5      retryCnt = 0; // Number of times we have sent the packet unsuccessfully.
6  5.2.2 REQ Arrival (Scheduler, Hardware)
7      if (sidQCtrl.sidCfgBits.arq == FALSE) {
8          scWinOff = REQ.winOff; // Scheduler
9          reqWinOff = REQ.reqWinOff; // Scheduler }
10 5.2.3 MAP Construction (Scheduler)
11     // Clear ErrorRecovery state for each new frame
12     state = normal;
13     while (Space left in current Upstream TDD frame) {
14         sidQCtrl = SID that Scheduler selects;
15         bytesInQueueToSchedule = reqWinOff - scWinOff;
16         // Always try to schedule bytes for SIDs without ARQ.
17         // For SIDs with ARQ, we need to make sure that we have not
18         // exhausted our window before we try to schedule some bytes.
19         if ( (sidQCtrl.sidCfgBits.arq == FALSE) OR
20             ((scWinOff + BytesScheduled - curWinOff) < ARQWindowSize) ) {
21             Allocate ticks for WPDU in upstream portion of TDD frame;
22             DATA_GRANT_IE.payloadSize = BytesScheduled;
23             DATA_GRANT_IE.winOff = scWinOff;
24             scWinOff = scWinOff + DATA_GRANT_IE.payloadSize; } }
25 5.2.4 WPDU Arrival (Hardware)
26     // Discard any bad wpdus
27     if (CRC Error) {
28         discard(WPDU);
29         discard any packet currently being reassembled; }
30     // If ARQ, discard any out of sequence wpdus
31     else if (sidQCtrl.sidCfgBits.arq == TRUE) AND (WPDU.winOff != curWinOff) ) {
32         Discard(WPDU);
33         Hw writes burst status to Fifo;
34         send bad or dropped status to Scheduler; }
35     // Good wpdu
36     else {
37         curWinOff = WPDU.winOff + WPDU.payloadSize;
38         // Discard packet cases -
39         // if a new packet arrives and we were previously assembling a packet
40         // if the packet arrives with an incorrect length
41         if ((WPDU.catPtr == 0) and (curPtr != curElPtr.pktPtr)) {
42             Discard(Partial assembled packet);
43             curPtr = 0;
44             curElPtr = NULL; }

```

```

1      // if possible, check the new packet for incorrect length
2      if (curElPtr.length != (curPktPtr - curElPtr.pktPtr))
3          discard packet;
4      send good status to Scheduler; }
5  5.2.5 WPDU Status Arrives (Scheduler)
6      if (wpdu good) AND ((sidQCtrl.sidCfgBits.arq == FALSE) OR
7          (ackWinOff == WPDU.winOff)) {
8          retryCnt = 0;
9          reqWinOff = WPDU.reqWinOff;
10         ackWinOff = WPDU.winOff + WPDU.length;
11         // WMAC Driver needs to calculate this and send the new winOff to the Scheduler
12         if (sidQCtrl.sidCfgBits.arq == FALSE)
13             scWinOff = WPDU.winOff + WPDU.length; }
14     else if ( ((wpdu lost) OR (wpdu bad)) AND (sidQCtrl.sidCfgBits.arq == TRUE) ) {
15         // By checking the state for ErrorRecovery, this means that we just
16         // reset the Scheduler's window for the first bad WPDU in the frame.
17         // The state is reset to normal during upstream map construction.
18         // If a MAP is lost, then the wpdus will be lost.
19         if ( (time > ErrorRecoveryTime) AND (retryCnt <= sidQCtrl.maxRetry) ) {
20             nakFlag = TRUE;
21             ErrorRecoveryTime = Tick Count at end of last scheduled upstream frame;
22             scWinOff = curWinOff;
23             // Should ackWinOff = curWinOff? Verify.
24             update sc pointers to cur pointers;
25             retryCnt = retryCnt + 1; } }
26  5.2.6 Flush Packet (Scheduler, Policer)
27      // When the retry count expires, drop the packet being assembled.
28      if (retryCnt > maxRetry) // Scheduler {
29          // When a packet is dropped, the CPE must make a new request.
30          scWinOff = curWinOff; // Scheduler
31          Update sc pointers to cur pointers; // Scheduler
32          retryCnt = 0; // Scheduler
33          flushFlag = 1; // Scheduler
34          Send msg to Link Adaptation Routine; // Scheduler
35          send msg to Policer with sidNum; // Scheduler
36          reqWinOff = curWinOff; }
37  5.2.7 Build MAP ACK IE Types (Scheduler)
38      if (sidQCtrl.sidCfgBits.arq == TRUE) {
39          if (nakFlag) {
40              NACK.sidNumber = sid;
41              NACK.winOff = curWinOff;
42              Put NACK in MAP;
43              nakflag = FALSE; }
44          else if (flushFlag) {

```

```
1      FLUSH.sidNumber = sid;
2      FLUSH.winOff = curWinOff;
3      Put FLUSH in MAP;
4      flushflag = 0; }
5  else {
6      ACK.sidNumber = sid;
7      ACK.winOff = ackWinOff;
8      Put ACK in MAP; } }
```